Attorney's Docket No. LS/0005.01

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Examiner: **Lin, Kelvin Y.**
Timothy W. GENSKE, William G. SWINTON, David VOGEL, Philippe R. KHAN, and **Eric O. BODNAR**	Art Group: **2142**
Application No. 09/847,811	
Filed: May 1, 2001	
For: **SYSTEM AND METHOD FOR DYNAMIC UPLOADING AND EXECUTION OF APPLICATIONS AND DRIVERS BETWEEN DEVICES**	

Mail Stop Appeal Briefs - Patent
Assistant Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Pursuant to 37 C.F.R. § 1.192, Applicants submit the following Appeal Brief for consideration by the Board of Patent Appeals and Interferences ("Board"). Applicants also submit herewith a check in the amount of $250.00 to cover the cost of filing this opening brief, as set forth in 37 C.F.R. § 1.17(c). Please charge any additional amounts due or credit any overpayment to Deposit Account No. 02-2666.

12/12/2005 MGEBREM1 00000035 09847811
02 FC:2402 250.00 OP

TABLE OF CONTENTS

I. Real Party in Interest 3
II. Related Appeals and Interferences 3
III. Status of Claims 3
IV. Status of Amendments 3
V. Summary of Invention 3
VI. Argument 4
A. Overview of Cited References 4
1. U.S. Patent No. 6,583,813 to Enright et al. ("Enright") 4
2. U.S. Patent Application No. 2002/0032027 by Kirani et al. ("Kirani") 5
B. No Reference Teaches Transmitting or Executing an Executable File 6
C. Other Rejections Are Supported By Random, Irrelevant Citations 9
D. Supplemental Reference Fails to Cure Defects of Primary 9
E. Rejections of XML Streams Are Unsupportable 10
VII. Conclusion 11
VIII. Claims Appendix 12
IX. Evidence Appendix 24
X. Related Proceedings Appendix 24

I. REAL PARTY IN INTEREST

LightSurf Technologies Inc. of Santa Cruz, California is the is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences that will directly affect, be directly affected by, or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-87 are pending in this application. All claims stand rejected. Claims 1-87 are presented for appeal.

IV. STATUS OF AMENDMENTS

Claims 1-3, 5-40 and 42-87 are as originally presented. Claim 41 was previously amended to correct a typographical error, and that amendment was entered. Applicants proposed an amendment to claim 4 that was believed to point out more particularly the material Applicants regard as their invention, but the Examiner declined to enter the amendment. Accordingly, claim 4 is presented in its original form in this appeal.

Applicants seek review of all rejected claims and ask the Board to overturn the Examiner's rejections based on arguments presented in support of independent claims 1, 41 and 51.

V. SUMMARY OF INVENTION

The invention concerns methodologies for uploading and executing applications and drivers between devices, where a client device probes its environment to identify a host to which it is connected (*see* Abstract; Summary p. 6, ll. 4-6; p. 23, ll. 10-12), then sends (uploads, transmits, injects) an executable driver or application to be executed by the host *(see* Abstract; p. 6, ll. 6-9; p. 23, ll. 12-15). The client invokes execution of the just-uploaded driver or application on the host *(see* Abstract; p. 7, ll. 11-15; p. 39, ll. 17-

19), and finally, waits for commands and interactions from the host (*see* Abstract; p. 23, ll. 15-19; p. 43, ll. 2-6).

The specific example of a digital camera being connected to a personal computer ("PC"), uploading a driver to the PC, and responding to commands from the driver is considered extensively as a preferred embodiment (pp. 10-23), including details such as communication protocols that could be used (*e.g.* TCP/IP, p. 24, ll. 10-23) and possible syntaxes for exchanging executable objects, commands and data (*e.g.* Extensible Markup Language or "XML," p. 26, ll. 10-24).

VI. ARGUMENT

The Examiner has rejected claims 1-12, 16-20, 22-27 and 29-87 under 35 U.S.C. § 102(e) as anticipated by U.S. Patent No. 6,583,813 to Enright *et al.* ("*Enright*"); and claims 13-15, 21 and 29 under 35 U.S.C. § 103(a) as unpatentable over *Enright* in view of U.S. Patent Application No. 2002/0032027 by Kirani *et al.* ("*Kirani*"). Applicants will present brief overviews of these references, then explain why the references are inadequate to support the Examiner's position and why, consequently, the Board should overturn the Examiner's determination and hold that the claims presented are patentable over the prior art of record.

A. Overview of Cited References

1. U.S. Patent No. 6,583,813 to Enright *et al.* ("*Enright*")

The primary reference relied upon by the Examiner is a U.S. patent granted to Enright *et al.* for a system and method for capturing and searching image data associated with transactions at automated teller machines ("ATMs") and similar terminals. Enright's system captures and stores images according to programmed sequences, and may also capture and store images in response to an alarm condition or motion detected in a video image. (*see* Abstract). Enright's design is concerned with recording images of documents and transactions at automated banking machines (*see* c. 3, ll. 30-32 and 65-67) and viewing and searching these recorded images (*see* c. 3, ll. 43-45; c. 4, ll. 37-39). Embodiments of Enright's system combine cameras, computers,

communication networks, and data storage servers (*see*, *e.g.*, Fig. 11) to record and play the images.

Enright discusses how a wide range of devices can interact with the system (*see*, *e.g.*, c. 10, ll. 63-66: card reader and keypad; c. 11, ll. 21-46: cameras; c. 12, ll. 3-9: image recorder; c. 13, ll. 2-5: pagers and cellular phones; etc.) However, *Enright* treats these components and their interactions at a higher level of abstraction than is relevant to Applicants' claimed invention, so it is a significant challenge to establish a concordance between *Enright*'s teachings and the elements of Applicants' claims, particularly considering the standard required to support a § 102 rejection. Two elements that are common to the rejected independent claims and clearly missing from *Enright* are the transmitting of an executable file from client to host, and invoking the execution of that file on the host.

2. U.S. Patent Application No. 2002/0032027 by Kirani *et al.* ("*Kirani*")

The Examiner relies on *Kirani* as a secondary reference in the rejections under 35 U.S.C. § 103(a) of several claims that depend directly or indirectly upon claim 1. Kirani's application describes a media spooler system and methodology to provide efficient transmission of media content from wireless devices. The media spooler acts as a protocol gateway between thin-client devices such as a wireless digital camera and a server based computer system of a photographic service provider. (*See* Abstract.) The gateway addresses transmission problems such as dropped cellular calls that may be encountered when uploading media content from client devices to a target host or server (*see* [0018]). The client device places a data call and establishes a TCP/IP connection to the media spooler, then transmits its information (for example, a digital photograph) (*see* [0195]). If the call is interrupted, data transfer can be resumed when a new connection is established (*see* [0196] – [0197]).

Kirani is only relied upon to establish that a wireless link could be replaced by a serial cable, but even assuming, *arguendo*, that *Kirani* establishes what the Examiner asserts, and that the references may properly be combined, the fact remains that neither *Enright* nor *Kirani* individually, nor the two together, teach or suggest transmitting an executable file from a client to its host, then invoking the execution of the file on the host.

B. No Reference Teaches Transmitting or Executing an Executable File

Independent claim 1 recites a method for automated transmission and execution of an executable file of interest originating from a digital camera, upon the camera's connection to a cellular phone, comprising a number of operations, including transmitting an executable file of interest from said camera to the cellular phone, and invoking execution of the executable file of interest after it has been transmitted to the cellular phone.

The Examiner asserts that these operations are taught at *Enright* c. 21, ll. 45-52 and c. 29, ll. 15-23. However, review of the cited sections fails to disclose any mention of an executable file, let alone transmitting such a file from a camera to a cellular phone and then invoking execution of the file. Instead, *Enright* at c. 21, ll. 45-52 discusses verifying a document inserted into an ATM by capturing and transmitting an image of the document while the transaction is ongoing. This is in the broader context of deciding what images to capture during a transaction, when (and under what circumstances) to capture them, and where to send the images and/or related messages (*see*, generally, *Enright* c. 20, l. 26 through c. 21, l. 59 and Figure 7). In the material the Examiner cites for support here, the sole mention of execution is of "programmable instructions executed in connection with image server 182," (*see* c. 29, ll. 20-21) but these instructions are not transmitted from a camera to a cellular phone, nor is their execution invoked after such transmission. Instead, the phrase "programmable instructions" seems simply to refer to the software of image server 182 (*see* Figure 11), which is neither a camera nor a cellular phone; the method by which the software arrives at the image server is never discussed.

Applicants advanced similar arguments in their Response to the Office Action mailed September 7, 2004, and the Examiner replied that *Enright* discloses an image download sequence, executed at the image server, to download an image to a remote terminal, which may be a cellular phone. The Examiner was referring to *Enright* at c. 33, ll. 30-32 and c. 13, l. 5.

However, even assuming that the "image download sequence" is an "executable file" within the meaning of claim 1, in *Enright,* the sequence is not transmitted *to* the image server, nor is its execution invoked *by* the cellular phone or remote terminal. The

cited material ("[i]f so, an image download sequence is executed at a step 266") is merely the consequence of a prior determination, "[whether] there is a concern about lack of memory." In other words, *Enright*'s image server may execute an image download sequence if the server is running out of memory, and the sequence will download images to a remote terminal or to a hard or soft permanent or temporary storage device.

Applicants have carefully reviewed *Enright*'s complete disclosure, with particular attention to the portions relied upon by the Examiner, and can find no suggestion of the claimed operation of transmitting an executable file from a camera to a cellular phone and invoking execution of the executable file after it has been transmitted to the cellular phone.

Support for the other aspects of the Examiner's rejection of claim 1 is also weak. For example, the passage:

> [The image server's connection to the network] enables the image recorder device to communicate with other types of remote terminals including terminals **connect**ed to wireless interfaces such as pagers and **cellular phone**s.
>
> *Enright* at c. 13, ll. 2-5

is alleged to anticipate the specific claimed operation of:

> **connect**ing the digital camera to a **cellular** tele**phone** capable of hosting the camera.

Certainly, one might infer that the *ability* to connect implies the specific act of doing so on some occasion, but even so, *Enright*'s image recorder is not a digital camera – instead, it is *connected to* some cameras, which provide (as *Enright* helpfully mentions) *analog* signals (*see* Figures 1, 11 and 12, and c. 12, ll. 17-18). There are simply too many gaps in the Examiner's analysis to support a rejection under 35 U.S.C. § 102(e) over *Enright*. Therefore, the Board should **overturn** the Examiner's rejection of claim 1 and allow that claim, and its dependent claims.

Independent claim 51 recites a method similar to that of claim 1, the method to be performed upon a first device's connection to a host device, and comprising transmitting an executable file of interest from the first device to the host device, and

transmitting from the first device to the host device commands that manipulate the executable file of interest at the host device. The Examiner omitted specific consideration of this claim, asserting only that "claims 51-67 have similar limitation as claims 1-2, 4, 6, 8, 31-32, 34-35, 38-39" and rejecting them for the same reasons. Even allowing that the broader claim terms "first device" and "host device" can be identified with components in *Enright*'s system as the Examiner sees fit, the fact remains that no two of *Enright*'s components interact as claim 51 recites. *Enright* does not teach or suggest a system in which the first device transmits an executable file to the host device, and then transmits commands to manipulate the executable file at the host device. As discussed above, *Enright* does not teach or suggest such interaction between the elements. Therefore, the Examiner has failed to establish a *prima facie* case that *Enright* anticipates claim 51 by teaching or suggesting each and every element of that claim, and the Board should **overturn** this rejection.

Independent claim 41 is drawn to a system for providing automated loading and execution of a driver required for connected devices comprising several elements, including a subsystem in a camera for automatically identifying a cellular phone upon its connection to the camera, uploading a driver from the camera to the phone, and transmitting at least one command from the camera that invokes execution of the driver at the phone. The Examiner omitted specific consideration of this claim as well, asserting that the limitations are similar to those of claim 1 and rejecting the claim for the same reasons. Applicants respectfully disagree. The limitations in this claim including (i) identifying the cellular phone upon connection to the camera and initiating communication between the two devices; (ii) uploading the driver of interest from the camera to the cellular phone; and (iii) transmitting at least one command from the camera that invokes execution of the driver of interest "identifying a cellular phone," are not taught or suggested by *Enright.* As discussed above, *Enright* does not teach or suggest the uploading of an executable file, such as a driver. It follows, therefore, that a subsystem to perform the operations cannot be anticipated either, so the Board should **overturn** this rejection.

C. Other Rejections Are Not Appropriate over the References

The Examiner's analysis and rejection of claims 2-12, 16-20, 22-27 and 29-40 is supported by citations to various portions of *Enright*, however, Applicants cannot find support in the reference for these rejections.

For example, claim 4 refines the method of claim 1, requiring that the executable file comprise a binary file having instructions capable of executing at the cellular phone. This is rejected over a citation to *Enright* c. 35, ll. 27-30 and 41-61, which discusses sending electronic mail to a plurality of individuals in response to the occurrence of a single event or other activity, and information that may be contained in the email message. Other rejections are equally baffling (*e.g.* claim 19, requiring that a default communication medium be selected by factory-preset information, allegedly anticipated by a brief discussion of time data and clock synchronization [c. 22, ll. 37-39]; or claim 24, reciting details of probing the identity of the cellular phone connected to the camera, rejected over actions taken when a sensor detects that a service door of an ATM is opened [c. 19, ll. 53-56] and the use of sound levels at an ATM [c. 39, ll. 39-40] to evaluate conditions there).

Each dependent claim should be held allowable by virtue of its dependence upon an allowable base claim, and also independently because these rejections simply do not anticipate the limitations in the dependent claims.

D. Supplemental Reference Fails to Cure Defects of Primary

The Examiner also rejected claims 13-15, 21 and 28 under 35 U.S.C. § 103(a) as unpatentable over *Enright* in view of *Kirani*. The latter reference is only relied upon for minor points: that a wireless interface might be replaced by a serial or USB cable, and that the Point-to-Point Protocol ("PPP") can be used to establish internet connectivity. However, Kirani does not teach or suggest uploading an executable file from a camera to a cellular phone and transmitting at least one command from the camera that invokes execution of the executable file on the host device Therefore, Kirani does not overcome the shortcomings of Enright discussed above. The Board should **overturn** the rejections of claims 13-15, 21 and 28.

E. Rejections of XML Stream Claims over Enright Should be Reversed

Claims 68-87 depend directly or indirectly upon claim 51, discussed above, and recite specific Extensible Markup Language ("XML") streams to carry the commands and data pertaining to the dialog claimed. The Examiner rejected those claims as anticipated by *Enright* because *Enright* mentions XML among markup languages that could be used in communicating Hypertext Transfer Protocol ("HTTP") messages (*see Enright*, c. 12, l. 59). In response to Applicants' observation that no specific XML streams are described in *Enright*, the Examiner replied "XML lets Web developers and designers created [sic] customized tags that offer greater flexibility in this area of skill. [Citation.] In addition, XML intends to be systemically arbitrary to perform identical functionality. Therefore, claims 68-87 are unpatentable." However, Applicants are not claiming XML itself, but rather the use of XML streams to carry commands and data pertaining to the dialog claimed. The Examiner does not make reference to any prior art which teaches or suggests such a dialog using XML streams.

Therefore, the Board is respectfully requested to **overturn** the rejections of claims 68-87.

VII. CONCLUSION

Based on the foregoing, Applicants respectfully submit that that the Board should overturn the rejection of all pending claims and hold that all of the claims currently under review are allowable.

Respectfully submitted,
Blakely, Sokoloff, Taylor & Zafman, LLP

Dated: December 8, 2005

William Thomas Babbitt, Reg. No. 39,591

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025

(310) 207-3800

CERTIFICATE OF MAILING

I hereby certify that the correspondence is being deposited with the United States Postal Service as first class mail in an envelope with sufficient postage, addressed to:

Mail Stop Appeal Brief - Patents
Commissioner for Patents
Alexandria, VA 22313-1450

Marilyn Bass — 12-8-05 Date

VIII. CLAIMS APPENDIX

The claims involved in this appeal are presented below.

1. (Original) In a computer environment where devices are occasionally connected together, a method for automated transmission and execution of an executable file of interest originating from a digital camera, upon the digital camera's connection to a cellular phone, the method comprising:

connecting the digital camera to a cellular phone capable of hosting the camera;

identifying at least one particular cellular phone that is connected to the camera, including determining communication information allowing communication between the camera and the particular cellular phone, and determining command information allowing the camera to invoke execution of a file of interest at the particular cellular phone;

based on said determined communication information, transmitting the executable file of interest from said camera to the particular cellular phone; and

based on said determined command information, invoking execution of the executable file of interest after it has been transmitted to the particular cellular phone.

2. (Original) The method of claim 1, wherein said executable file of interest comprises a driver file.

3. (Original) The method of claim 2, wherein said driver file, upon execution, controls operation of said camera.

4. (Original) The method of claim 1, wherein said executable file comprises a binary file having instructions capable of executing at said cellular phone.

5. (Original) The method of claim 1, wherein said executable file comprises an application program capable of executing at said cellular phone.

6. (Original) The method of claim 1, wherein said camera includes an add-in device capable of being hosted by said cellular phone.

7. (Original) The method of claim 6, wherein said camera comprises a digital camera and wherein said method further comprises:

upon execution of said executable file at said cellular phone, transferring image information from said digital camera to said cellular phone.

8. (Original) At the method of claim 7, further comprising:

after transferring said image information from said digital camera to said cellular phone, wirelessly transmitting said image information to a third device.

9. (Original) The method of claim 1, wherein said cellular phone includes a computing device capable of hosting other devices.

10. (Original) The method of claim 1, wherein said cellular phone includes wireless transmission capability for transferring information received from said camera to other devices.

11. (Original) The method of claim 1, wherein said camera and cellular phones are occasionally connected together.

12. (Original) The method of claim 1, wherein said camera and cellular phones are permanently connected together.

13. (Original) The method of claim 1, wherein said camera and cellular phones are connected together via a serial communication link.

14. (Original) The method of claim 13, wherein said serial communication link comprises an RS-232 serial communication link.

15. (Original) The method of claim 1, wherein said camera and cellular phones are connected together via a USB (Universal Serial Bus) link.

16. (Original) The method of claim 1, wherein invocation of said identifying step occurs upon connecting said camera and cellular phones together.

17. (Original) The method of claim 1, wherein said identifying step includes:

probing the camera's environment for determining which devices, if any, the camera is attached to.

18. (Original) The method of claim 17, wherein said probing step includes:

determining a default communication medium for probing for new devices.

19. (Original) The method of claim 18, wherein said default communication medium is specified initially by factory-preset information.

20. (Original) The method of claim 18, wherein said default communication medium is a selected one of a wireless and a wired communication medium.

21. (Original) The method of claim 20, wherein said default communication medium includes a serial (RS-232) and a USB (Universal Serial Bus) wired communication medium.

22. (Original) The method of claim 19, wherein said factory-preset information is stored in a registry of the camera.

23. (Original) The method of claim 19, wherein said factory-preset information includes a default communication rate and default handshake protocol for at least one potential cellular phone.

24. (Original) The method of claim 17, wherein said probing step includes:

executing an initial sequence of handshake commands and comparing any response received to a list of known responses for identifying a particular cellular phone.

25. (Original) The method of claim 17, wherein said probing step continues until all known potential cellular phones have been enumerated.

26. (Original) The method of claim 1, wherein said identifying step includes:

updating a registry at said camera for indicating any connected cellular phone that has been identified.

27. (Original) The method of claim 1, further comprising:

upon identifying at least one particular cellular phone, ensuring that a state of TCP/IP communication is reached between said camera and the particular identified cellular phone.

28. (Original) The method of claim 27, wherein said step of ensuring that a state of TCP/IP communication is reached includes:

initiating a PPP (Point-to-Point Protocol) communication session between said camera and cellular phones; and, thereafter

initiating a TCP/IP communication session between said camera and cellular phones.

29. (Original) The method of claim 27, wherein said step of ensuring that a state of TCP/IP communication is reached includes:

determining an IP (Internet Protocol) address for said cellular phone.

30. (Original) The method of claim 1, wherein said step of transmitting the executable file of interest includes:

opening the executable file of interest at the camera; and

streaming the opened executable file of interest from the camera to the cellular phone.

31. (Original) The method of claim 30, wherein said streaming step includes:

employing XML protocol for packaging said executable file of interest for delivery to the cellular phone.

32. (Original) The method of claim 30, wherein said step of transmitting further comprises:

returning to said camera a file handle permitting said camera to access said executable file of interest transmitted to said cellular phone.

33. (Original) The method of claim 31, wherein said file handle comprises a file handle that may be understood by said cellular phone for accessing a particular file of interest at said cellular phone.

34. (Original) The method of claim 1, wherein said executable file of interest comprises a byte-code program, and wherein said cellular phone includes capability for executing byte-code programs.

35. (Original) The method of claim 1, wherein said executable file of interest comprises a Java program, and wherein said cellular phone includes a Java Virtual Machine for executing Java programs.

36. (Original) The method of claim 1, wherein said step of invoking execution of the executable file of interest includes:

issuing a command from said camera to said cellular phone to begin execution at said cellular phone of said executable file of interest.

37. (Original) The method of claim 1, wherein said step of invoking execution of the executable file of interest includes:

triggering execution of said executable file indirectly at said cellular phone by instructing said cellular phone to restart itself.

38. (Original) The method of claim 1, further comprising:

placing said camera in a listening mode, after said camera has invoked execution of said executable file at said cellular phone.

39. (Original) The method of claim 38, wherein said camera awaits commands from said cellular phone, while said camera is in a listening mode.

40. (Original) The method of claim 39, wherein commands received at said camera from said cellular phone control operation of said camera.

41. (Previously Presented) A multi-device system providing automated loading and execution of a driver required for connected devices, the system comprising:

a camera that may be connected to a cellular phone that is capable of hosting the camera; and

a subsystem, incorporated in the camera, for automatically:

(i) identifying the cellular phone upon connection to the camera, said subsystem initiating communication between the two devices;

(ii) uploading the driver of interest from the camera to the cellular phone; and

(iii) transmitting at least one command from the camera that invokes execution of the driver of interest at the cellular phone, whereupon the driver executes at the cellular phone for controlling operation of the camera.

42. (Original) The system of claim 41, wherein said driver comprises a binary file having instructions capable of executing at said cellular phone.

43. (Original) The system of claim 42, wherein said binary file comprises native machine instructions for execution by a processor at said cellular phone.

44. (Original) The system of claim 42, wherein said binary file comprises byte-code instructions for execution by an interpreter at said cellular phone.

45. (Original) The system of claim 44, wherein said binary file comprises a Java program, and wherein said cellular phone includes a Java Virtual Machine for executing Java programs.

46. (Original) The system of claim 44, wherein said driver includes:

instructions for unpacking other executable files for execution at said cellular phone.

47. (Original) The system of claim 41, wherein said camera comprises an add-in device capable of being hosted by said cellular phone.

48. (Original) The system of claim 47, wherein said camera comprises a digital camera device, and wherein said cellular phone comprises a handheld device capable of hosting said digital camera device.

49. (Original) The system of claim 48, wherein said handheld computing device functions to retrieve digital image information from said digital camera device and wirelessly transmit that information to another system.

50. (Original) The system of claim 48, wherein said handheld device is a selected one of a cellular phone device and a handheld computing device.

51. (Original) In a computer environment where devices are occasionally connected together, a method for automated transmission, execution, and manipulation of an executable file of interest originating from a first device, upon the first device's connection to a host device, the method comprising:

connecting the first device to at least one other device capable of hosting the first device;

identifying at least one particular host device that is connected to the first device, including determining communication information allowing communication between the first device and the particular host device, and determining command information allowing the first device to manipulate and invoke execution of an executable file of interest at the particular host device;

based on said determined communication information, transmitting the executable file of interest from said first device to the particular host device;

based on said determined command information, transmitting from said first device to the particular host device commands that manipulate the executable file of interest at the particular host device; and

initiating a dialog between the two devices, including:

(i) executing said commands transmitted to the host device on the host device, and

(ii) in response to said commands transmitted to the host device, returning a reply from the host device to the first device.

52. (Original) The method of claim 51, wherein said commands include a command to load the executable file of interest.

53. (Original) The method of claim 51, wherein said commands include a command to start the executable file of interest.

54. (Original) The method of claim 51, wherein said commands include a command to end the executable file of interest.

55. (Original) The method of claim 51, wherein said commands include a command to activate the executable file of interest.

56. (Original) The method of claim 51, wherein said commands include a command to get the capabilities of the host device.

57. (Original) The method of claim 51, wherein said commands include a command to get a reference to the executable file of interest that is running on the host device.

58. (Original) The method of claim 51, wherein said executable file of interest comprises a Java program, and wherein said host device includes a Java Virtual Machine for executing Java programs.

59. (Original) The method of claim 51, wherein said executable file of interest comprises a byte-code program, and wherein said host device includes capability for executing byte-code programs.

60. (Original) The method of claim 51, further comprising:

placing said host device in a listening mode to receive commands from said first device.

61. (Original) The method of claim 51, further comprising:

after said first device has transmitted a command to said host device, placing said first device in a listening mode to receive a reply transmitted from said host device.

62. (Original) The method of claim 51, wherein said reply transmitted from the host device in response to said command from the first device includes status information.

63. (Original) The method of claim 62, wherein said status information includes error information indicating an execution state of a preceding command executed at the host device.

64. (Original) The method of claim 51, wherein transmission between the devices employs XML protocol.

65. (Original) The method of claim 51, further comprising:

returning to said first device a file handle permitting said first device to access said executable file of interest while it resides at said host device.

66. (Original) The method of claim 65, wherein said file handle comprises a file handle that may be understood by said host device for accessing a particular file of interest at said host device.

67. (Original) The method of claim 51, wherein said dialog includes:

issuing a load application command from said first device to said host device to receive said executable file of interest transmitted from the first device.

68. (Original) The method of claim 67, wherein the load application command is transmitted from the first device to the host device as an XML stream with a syntax of:

```
<LoadApp>
    <name>(app)</name>
    <bin>
        <size>(value)</size>
        (data)
    </bin>
</LoadApp>
```

69. (Original) The method of claim 67, wherein the reply to the load application command is transmitted by the host device to the first device as an XML stream with a syntax of:

```
<LoadAppR>
    <status>(value)</status>
    <handle>(value)</handle>
</LoadAppR>
```

70. (Original) The method of claim 51, wherein said dialog includes:

issuing a release application command from said first device to said host device to be able to delete said executable file of interest.

71. (Original) The method of claim 70, wherein the release application command is transmitted from the first device to the host device as an XML stream with a syntax of:

```
<ReleaseApp>
    <handle>(value)</handle>
</ReleaseApp>
```

72. (Original) The method of claim 70, wherein a reply to the release application command is transmitted by the host device to the first device as an XML stream with a syntax of:

```
<ReleaseAppR>
    <status>(value)</status>
</ReleaseAppR>
```

73. (Original) The method of claim 51, wherein said dialog includes:

issuing a start application command from said first device to said host device to begin execution at said host device of said executable file of interest.

74. (Original) The method of claim 73, wherein the start application command is transmitted from the first device to the host device as an XML stream with a syntax of:

```
<StartApp>
    <handle>(value)</handle>  //Handle to application
</StartApp>
```

75. (Original) The method of claim 73,wherein the reply to the start application command is transmitted by the host device to the first device as an XML stream with the following syntax:

```
<StartAppR>
    <status>(value)</status>  //Standard error replies
</StartAppR>
```

76. (Original) The method of claim 51, wherein said dialog includes:

issuing a stop application command from said first device to said host device to discontinue execution at said host device of said executable file of interest.

77. (Original) The method of claim 76, wherein the stop application command is transmitted from the first device to the host device as an XML stream with a syntax of:

```
<StopApp>
    <handle>(value)</handle>
    <priority>(value)</priority>
</StopApp>
```

78. (Original) The method of claim 76, wherein the reply to the stop application command is transmitted by the host device to the first device as an XML stream with a syntax of:

```
<StopAppR>
    <status>(value)</status>
</StopAppR>
```

79. (Original) The method of claim 51, wherein said dialog includes:

issuing an activate application command from said first device to said host device to bring current execution of said executable file of interest to the forefront at said host device.

80. (Original) The method of claim 79, wherein the activate application command is transmitted from the first device to the host device as an XML stream with a syntax of:

```
<ActivateApp>
    <handle>(value)</handle>
    <priority>(value)</priority>
</ActivateApp>
```

81. (Original) The method of claim 79, wherein a reply to the activate application command is transmitted by the host device to the first device as an XML stream with a syntax of:

```
<ActivateAppR>
    <status>(value)</status>
</ActivateAppR>
```

82. (Original) The method of claim 51, wherein said dialog includes:

issuing a command to get information about device capabilities of said host device.

83. (Original) The method of claim 82, wherein the device capabilities command is transmitted from the first device to the host device as an XML stream with a syntax of:

```
<GetCap>
</GetCap>
```

84. (Original) The method of claim 82, wherein the reply to the get capabilities command is transmitted by the host device to the first device as an XML stream with a syntax of:

```
<GetCapR>
    <status>(value)</status>
    <lang>(value)</lang>
    <id>(value)</id>
    <imei>(value)</imei>
    <imsi>(value)</imsi>
    <screen>(value)</screen>
    <version>(value)</version>
    <dataLink>(value)</dataLink>
    <flash>(value)</flash>
    <cpu>(value)</cpu>
</GetCapR>
```

85. (Original) The method of claim 51, wherein said dialog includes:

issuing a command to get information about an active application handle for the executable file of interest.

86. (Original) The method of claim 85, wherein the command to get information about an active application handle is transmitted from the first device to the host device as an XML stream with a syntax of:

```
<GetActAppHandle>
</GetActAppHandle>
```

87. (Original) The method of claim 85, wherein a reply to the command to get information about an active application handle is transmitted by the host device to the first device as an XML stream with a syntax of:

```
<GetActAppHandleR>
    <status>(value)</status>
    <handle>(value)</handle>
</GetActAppHandleR>
```

IX. EVIDENCE APPENDIX

No other evidence is submitted in connection with this appeal.

X. RELATED PROCEEDINGS APPENDIX

No related proceedings exist.